251900-68-F

FINAL REPORT

THE EXPERIMENT CONTROL SYSTEM
For The
ROBOT OPERATED MATERIAL PROCESSING SYSTEM (ROMPS) - MISSION 1

M. DOBBS

FEBRUARY 1995

This work performed under Contract NAS5-32471 for:

NASA Goddard Space Flight Center
Code 714
Greenbelt, MD 20771

ERIM P.O. Box 134001
Ann Arbor, MI 48113-4001

TECHNICAL REPORT STANDARD TITLE PAGE

1. Report No. 251900-68-F	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle The Experiment Control System for the Robot Operated Material Processing System (ROMPS) - Mission 1		5. Report Date February 1995
		6. Performing Organization Code
7. Author(s) Michael Dobbs		8. Performing Organization Report No. 251900-68-F
9. Performing Organization Name and Address ERIM P.O. Box 134001 Ann Arbor, MI 48113-4001		10. Work Unit No.
		11. Contract or Grant No. NAS5-32471
12. Sponsoring Agency Name and Address NASA Goddard Space Flight Center Code 714 Greenbelt, MD 20771		13. Type of Report and Period Covered Final Report
		14. Sponsoring Agency Code
15. Supplementary Notes		

16. Abstract

The Robot Operated Material Processing System (ROMPS) was designed to make available to the microgravity research community the same tools and mode of automated experimentation that their ground-based counterparts have enjoyed for the last two decades. This design goal was accomplished by combining commercial automation tools familiar to the experimenter community with system control components that interface with the on-orbit platform in a distributed architecture. This architecture insulates the experimenter from the details of the on-orbit platform while providing the payload engineer with the tools necessary for managing a payload. By using commercial software and hardware components whenever possible, development costs were greatly reduced when compared to traditional space development projects. Using commercial components also improved the usability of the system by providing familiar user interfaces, providing a wealth of readily available documentation, and reducing the need for training on system-specific details. The modularity of the distributed architecture implemented for ROMPS makes it very amenable for modification to other on-orbit experiments requiring robotics-based automation.

17. Key Words		18. Distribution Statement Unlimited Distribution	
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 3	22. Price NA

Executive Summary:

The Experiment Control System met 100% of the ROMPS mission primary (science) objectives and 100% of the secondary (engineering) objectives. The mission operated in the planned autonomous mode throughout, which enabled the completion of primary mission objectives, in approximately 1/3 of the allocated mission time. The secondary objectives, which required post integration development of flight scripts, demonstrated closed loop control of robotics via the Capaciflector system.

The ECS went from CDR to delivery in approximately 12 months. Integration of the ECS avionics with the GSFC robot was initially demonstrated within a few days. Integration with the ITE furnace took an additional week. Total integration, from start to finish, including significant effort towards the refinement of flight operations procedures, was completed in 3 months.

The ECS system performed as designed, with one minor anomaly reported in a low level EasyLab software module. Significantly, none of the three industrial grade processors experienced any radiation induced 'hits' during any portion of the mission, including significant time over the South Atlantic Anomaly.

Commercialization:

The Space Automation and Robotics Center of ERIM has licensed the Space Resources Division of Advanced Modular Power Systems (AMPS) to market the ECS technology. The commercial product ECS-II, is being actively marketed to potential users of GAS, Hitchhiker, WSF SmartCan, ELV's and Space Station Express platforms. A draft copy of the ECS-II Product Flyer is enclosed with this report.

Technical Discussion:

Spacecraft Command Language

Both the flight and ground SCL segments performed as designed. SCL's script upload capability was used to; a) reprogram operational sequences, b) obtain additional calibration data, c) operate with a changed STS mission profile, and d) work around a subsystem anomaly.

EasyLab

The EasyLab 'robot' procedures performed as designed. The operators did use the direct EasyLab command capability (SCL pass through mode) to diagnose two anomalies. The first was in the robot proper. Low level robot (joint) commands were sent in realtime to diagnose a hardware fault. The second was in a low level EasyLab module. Low level (furnace control) commands were sent to diagnose a software bug in a time delay routine.

XP Servo

Though primitive by some standards, the XP servo system not only performed flawlessly, but did not require any adjustments to the PID coefficients. The coefficients were developed by analysis and empirical means, in a 1-G environments. There was some concern that the coefficients might need to be changed for 0-G, but this did not occur.

Testbed(s)
While not part of the mission per se, the Zymark and ECS testbeds were used extensively during ECS hardware and software development. The testbeds were the critical component that enabled the rapid integration of the avionics and robotics at GSFC.

Ground Station
The ground station, composed of 100% commercial personal computers, performed as planned. An unexpected anomaly was uncovered in SCL's HH-DATAIO module. When the HH link was interrupted many, many times during a few second period, as happened during AOS/LOS, the ground segment SCL had a tendency to hang, which required rebooting. This rapid, recurring loss of the link was not tested by ERIM or ICS, nor was it part of the HH integration test program or mission simulations. No mission data was lost.

ROMPS-2 Improvements:

There are several industry/user-driven improvements that should be made in ROMPS to enhance its attractiveness to the private sector. Furthermore, the development effort required to accomplish these improvements can be directly applied to all material processing programs, specifically the WSF program.

- Non-contact temperature measurement
- Higher performance furnace - increased area, higher temperature and improved uniformity
- Database interface for telemetry reports

Investment Carryover (Return on Investment):

The NASA (GSFC and SpARC) investment in ROMPS has provided direct benefit to the RS3 and AWCS programs, and has stimulated the private sector (Advanced Modular Power Systems and Vision Instruments) to make additional investment in products for commercial space.

ROMPS robotics	AWCS robotics design, parts selection
Zymark software development	Vision Instruments Inc. products - Liquid Level Inspection Station, Automated Colony Transfer System
ROMPS ECS	AMPS Inc. product - ECS-II
RS3 upgrade of SCL for process control	WSF process control
ROMPS flight operations document	RS3 flight operations

Publications:

P. Olsztyn, M. Dobbs, G. Dickerhoof, and P. Dorrance, "A Space-based Laboratory Automation Architecture for Material Processing Experimentation," National Symposium on Laboratory Automation and Robotics Proceedings, October 17-20, 1993, Boston, MA.

ERIM

P. Olsztyn, D. Conrad, M. Dobbs, " A Distributed Architecture for On-Orbit Laboratory Automation and Robotics using COTS Components," 1994 AIAA 94-406 Space Programs and Technologies Conference

Conclusions:

The application of commercial technology provides significant advantages in terms of performance, cost and schedule. Commercial technology can be applied in the form of hardware and/or software, as components or systems. The risks of using 'non mil-spec' components can be quantified. Furthermore, we contend that the use of 'proven' software, versus newly developed, lowers overall programmatic risk.

The Experiment Control System was a successful experiment in the application of industrial electronics and commercial software. The ROMPS-1 mission clearly demonstrated the advantages in terms of cost and flexibility. This government sector seed investment, followed with private sector investment has lowered the cost of access to space and created high-technology, high-wage jobs.

Acknowledgments:

This work was performed for Mr. Lloyd Purves and Mr. Del Jenstrom of NASA Goddard Space Flight Center Code 714, under NASA contract NAS5-32471. Additional development funds were made available by Mr. David Conrad, Director of the Space Automation and Robotics Center, through NASA grant NAGW-1198.

As with any effort, there are many people that contributed to the success of the mission. The ECS team consisted of : G. Dickerhoof, P. Dorrance, J. Eder, E. Kwon, M. Massey, P. Olsztyn, R. Quada, N. Thomas, G. Wassick.

The ECS team would like to thank everyone at Interface & Control Systems, Zymark and ITE Inc. for their effort and dedication. I would like to especially thank three people: William Buote of Zymark for licensing ERIM to use Zymark's proprietary technology, Pat Cappelaere for supporting the Macintosh platform, and Ed Aaron of ITE for his sage advice.

A Spaced Based Laboratory Automation Architecture for Material Processing Experimentation

Paul Olsztyn Mike Dobbs
Greg Dickerhoof Pam Dorrance
Space Automation and Robotics Center
Ann Arbor, Michigan

Abstract
Developments in Laboratory Automation over the last 20 years have had a profound impact on the terrestrial analytical research and development process. Gains in both productivity and quality have been realized by the use of automated sample preparation, handling, measurement, and data analysis. Yet the space based researcher has recognized few benefits from these advances. The Robot Operated Material Processing System (ROMPS) was designed to make available to the microgravity research community the same tools and mode of automated experimentation that his ground based counterpart has enjoyed for the last two decades. To accomplish this design goal a survey of current terrestrial Laboratory Automation equipment and practices was conducted. The results of this study showed that existing Laboratory Automation technology could be modified for a space based platform of operation. The use of existing Laboratory Automation technology benefits the microgravity researcher by providing greater access to his experiment at a lower cost than the traditional space based experiment implementation.

Introduction
The SPace Automation and Robotics Center (SPARC) is a NASA sponsored Center for the Commercial Development of Space (CCDS). There are currently 17 CCDS located through out the United States all tasked with the broad charter to encourage U.S. private sector leadership in space related commerce. In addition to this broad charter, each CCDS has a set of more focussed objectives. At SpARC these objectives include the development of a space automation supplier industry. The economic viability of this industry is dependent upon the discovery of high value products whose manufacture relies upon the unique properties of space (microgravity, ultra-high vacuum, etc...). As discovery is the current focus of the space automation supplier industry, SpARC's first business objective is the reduction of the prohibitively high cost of space based research. Preliminary market research indicates that the extension of terrestrial laboratory automation techniques to the space environment could greatly reduce these costs while increasing the investigators ability to perform research. The Robot Operated Material Processing System (ROMPS) project provided SpARC with an opportunity to test the viability of these assumptions. This paper examines the system architecture of the ROMPS Experiment Control System which was built around Zymark's Zymate System V Controller and Interface and Control Systems' Spacecraft Command Language.

ROMPS Mission Overview
In 1991 SpARC was requested by NASA's Goddard Space Flight Center (GSFC) to propose an alternative Experiment Control System (ECS) design for the NASA Office of Commercial Programs project ROMPS. The ROMPS project is a Space Shuttle Hitchhiker mission centered around the Rapid Thermal Processing (RTP) of semiconductor materials. The ROMPS mission's short term objective are to develop commercially promising in-space processes by using a robot based automation system to lower the cost of the material processing procedures. Figure 1 shows the payload configuration of the ROMPS Hitchhiker Payload.

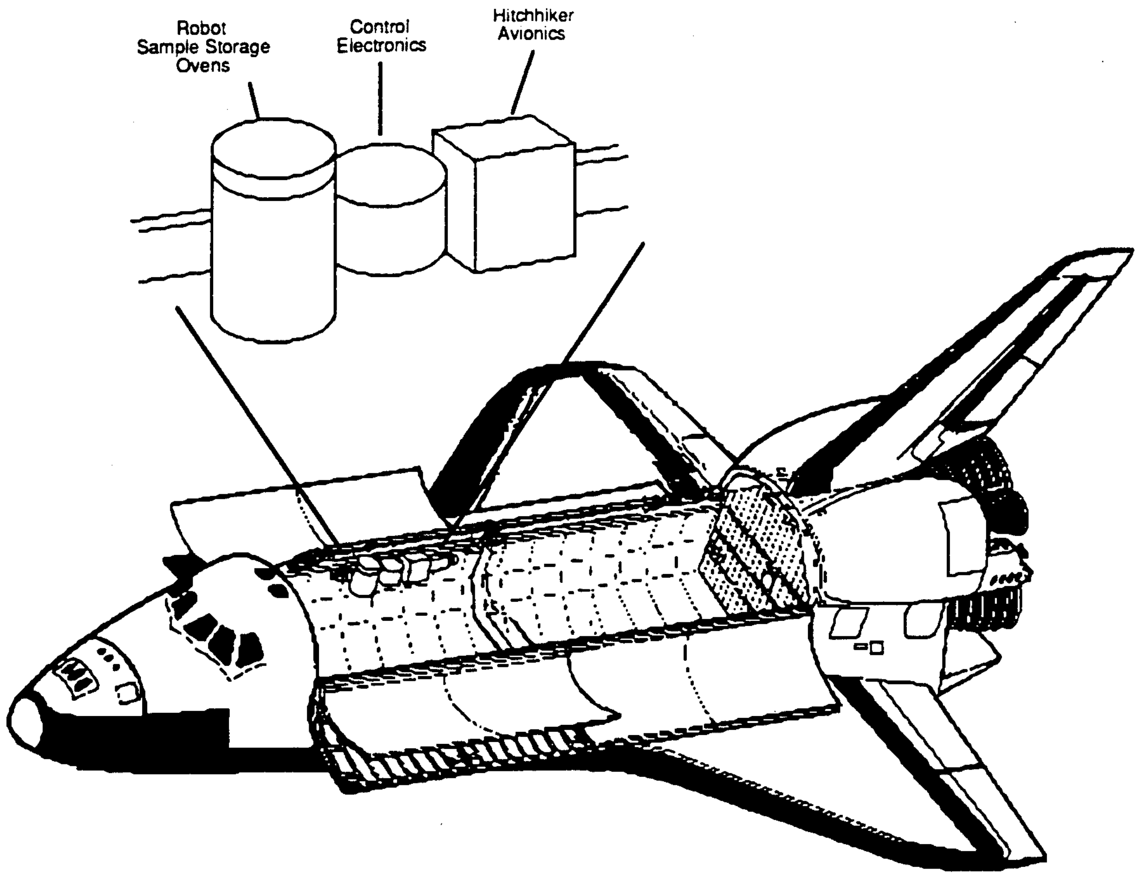


Figure 1 ROMPS Hitchhiker Payload Configuration

ROMPS Automated Rapid Thermal Processing

Rapid Thermal Processing is a widely used industrial material processing procedure for essentially 2-D semiconductor materials. In this process a heat source capable of producing uniform surface area temperatures (usually a quartz halogen lamp) is used to melt semiconductor materials. The semiconductor materials are then allowed to cool and recrystallize. The crystalline structure of the semiconductor materials determines many of their electrical properties. Gravity driven convection, buoyancy, and sedimentation during the recrystallization process of RTP influence the structure of the semiconductor grains produced. The semiconductor grains produced in microgravity have larger and more uniform crystals with better electrical properties. The possible applications for microgravity grown semiconductors include radiation hard micro-electronics, Gauss Meters, Watt Meters, and other optical and electronic devices[1].

ROMPS will provide a robotic based system capable of automating the RTP of a large number of samples, up to 155+. In this system a custom built Halogen Lamp Furnace will provide the heat source for the RTP of the samples. A robot designed by GSFC will move the semiconductor samples between their storage racks and the furnace. Figure 2 is a picture of the ROMPS robot, furnace, and storage racks under development at GSFC. SpARC's role in the development of ROMPS is to provide the computer and electrical subsystems for controlling and monitoring the ROMPS Payload.

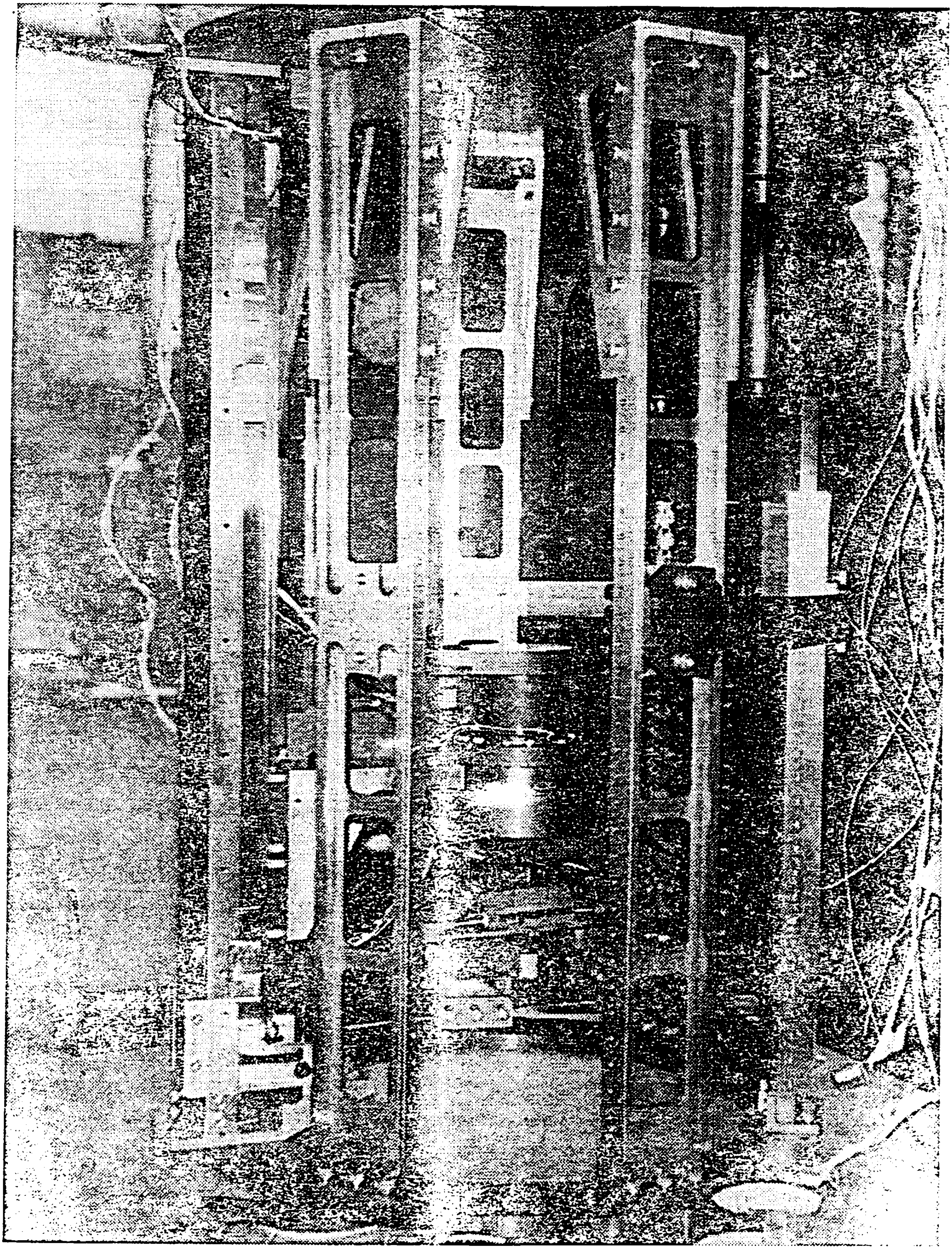


Figure 2 ROMPS Robot and Furnace under assembly at GSFC

ROMPS Mission Requirements Partitioning

As the first step our design process of the ROMPS Experiment Control System, requirements were gathered and Functional Partitioning performed. Functional Partitioning is a structured design methodology which allows the grouping of "like" functions in subsystem definitions without unnecessary influence from traditional subsystem boundaries[2]. A summary of this Functional Partitioning is shown below.

Robot Requirements
- Nominal Motion Control of Robot Arm/Gripper
- Robot protection for End of Travel and Over-force Conditions
- Robot Position Calibration
- Stored Robot Movement Sequences for Rack, Furnace, and Station Lock access
- Maintain/Report Robot Status/Sensor Data

Furnace Controller Requirements
- Set Furnace Controller Target Power/Temperature Setpoint
- Timed On/Off Control of Furnace Halogen Lamp
- Execution of a Seven Step Temperature/Time Profile

Payload Controller Requirements
- Experiment/Engineering Data Collection and Transmission
- Support Hitchhiker Asynchronous Uplink/Downlink
- Scripted control of RTP processing steps
- Monitoring of Experiment and Engineering Data with limited Safety Response
- Storage of Initial RTP Processing Parameters and Processing Schedules
- In-Flight modification of RTP Processing Parameters and Processing Schedules

Ground Station Requirements
- Man Machine Command/Telemetry Interface to Payload via Hitchhiker Customer Carrier Ground Support Equipment
- HH Bilevel Command Interface
- Start/Stop RTP scripted processing procedures
- Display/Monitor Telemetered Engineering and Experimental Data
- Archiving/Playback of Telemetered Data
- Upload/Ground Tracking of Experiment Parameter and Schedule Changes

Trade Study of Existing Automation and Robotic Systems

After Functional Partitioning, a trade study of existing data acquisition, control, robotics, and laboratory automation systems was conducted. This survey indicated that no single existing system could satisfy all the ROMPS mission requirements, but a combination of two or more commercial systems could be modified to meet them. Zymark's Zymate System V Controller was chosen for handling the Robot and Furnace Controller requirement partitions. This selection was based on the superiority of their robot control software, the high modularity of the System V Controller code, and their preeminent position in the field of laboratory automation. For the Payload Controller and Ground Station Requirements the Spacecraft Command Language (SCL) developed by Interface and Control Systems was selected. SCL provides a payload controller kernel which combines features of a Fifth Generation Programming Language, Expert System, and Distributed Database into a package well suited for the development of payload and ground station systems. In addition to the payload controller kernel, SCL provides a Macintosh development environment used to develop the scripts and rules executed by the SCL kernel. The development environment can serve as the base for a ground station command and

telemetry console. Another factor for the selection of the ICS and Zymark products as the base of our system was their willingness to work with SpARC not only as a sub-contractor, and product provider, but as a partner in the development process. Without this working relationship the ability to use Commercial Off The Shelf (COTS) software/hardware is reduced to "what you see is what you get".

ROMPS Distributed System Architecture

After the selection of the ICS and Zymark products as the core of the Experiment Control System, the process of laying out a hardware/software architecture and re-mapping the system requirements to specific subsystems began. In Figure 3 we see the migration path of the EasyLab System V Controller, Zymate XP Servo Controller, and ICS SCL Macintosh Development System from their native hardware architectures to a space hardened architecture which could meet the vibrational, thermal, and power requirements specified by the Shuttle Hitchhiker Customer Accommodation Requirements. This new architecture also provides interfaces for the engineering and experimental data collection sensors, and discrete command interfaces not needed in a terrestrial laboratory system.

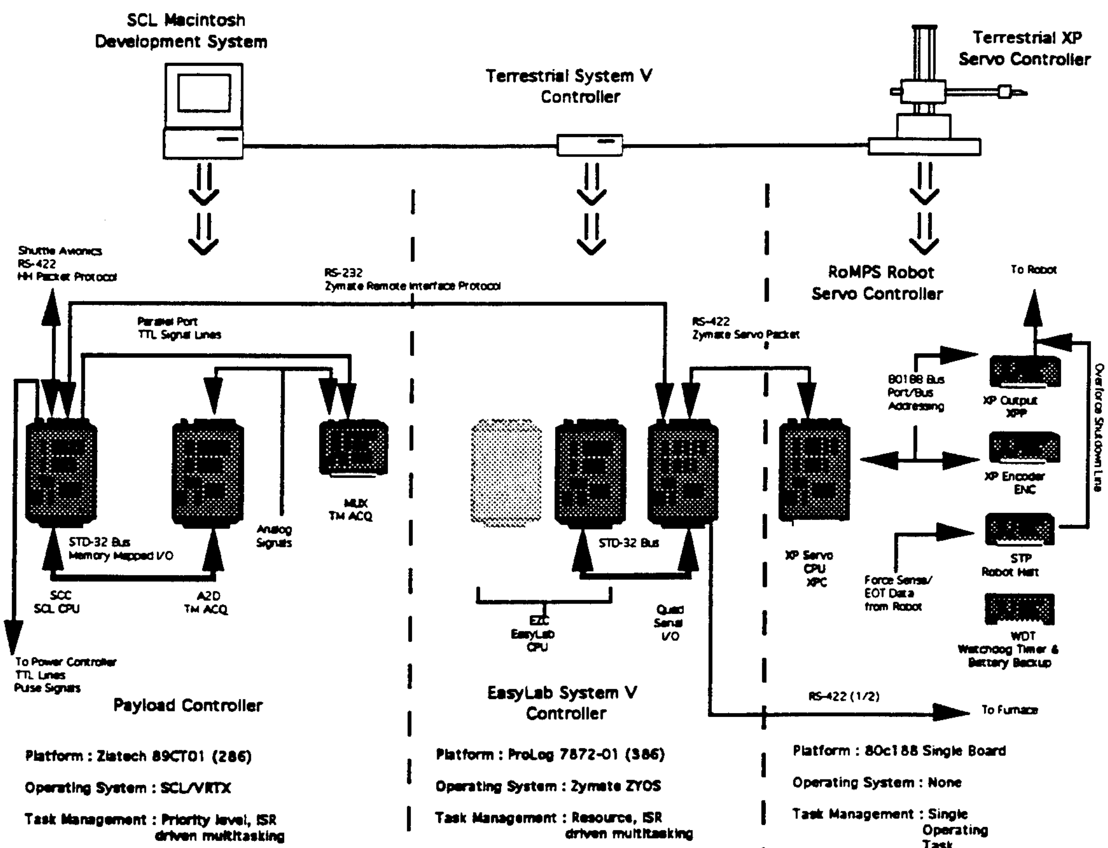
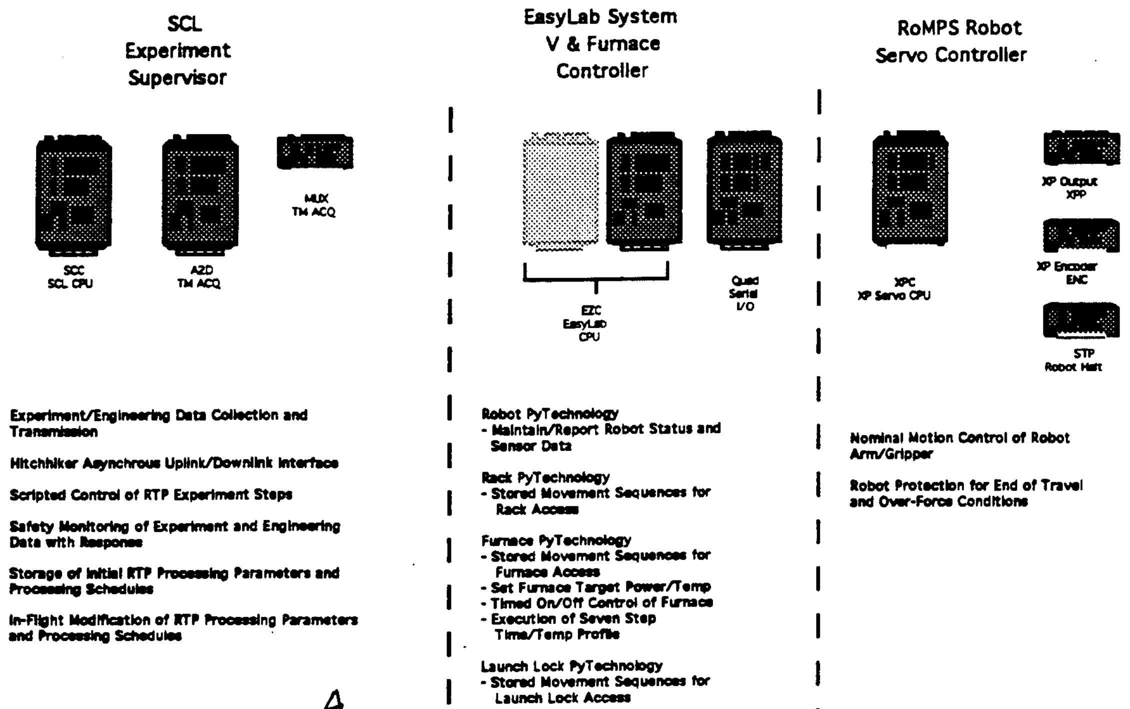


Figure 3 Migration of Terrestrial Architecture to a Space Hardened Architecture

The Flight Segment Architecture

Figure 4 summarizes the mapping of Functional Requirements onto the flight subsystems of the Experimental Control System. This figure also reveals the hierarchical levels of experiment control performed by the Experiment Control System's subsystems.



4
~~#3~~

> Figure ~~3~~ Requirements Mapping of Flight Subsystems

At the SCL Experiment Supervisor the control of the experiment is at a very high level. This subsystem executes the RTP processing scripts which sets Sample Processing Parameters and initiate Laboratory Unit Operations by sending commands to the Zymate System V Controller. This subsystem also monitors the health and safety of the payload, and halts the RTP script if the some anomalous condition is detected, such as temperature of the Furnace Structure exceeding some safe operating limit. The Zymate System V Controller executes the steps of the Laboratory Unit Operations initiated by SCL Experiment Supervisor. Device specific commands are sent to the ROMPS Robot Servo and Furnace Controllers to complete these Laboratory Unit Operations. At the ROMPS Robot Servo Controller the device specific robot commands sent from the EasyLab System V Controller are processed. This subsystem also protects the robot from harming itself during movement command sequences. Many of the disadvantages of a distributed architecture are eliminated in this system by the use of simple and robust communication protocols.

The Ground Segment Architecture

Figure 5 summarizes the mapping of Functional Requirements onto the ground station subsystems of the Experiment Control System. Again, the hierarchical levels of command and control are shown in this figure. Decreasing levels of experiment control abstraction are shown as one proceeds from left to right across this figure. At the far left the Experiment Display Console displays and prints those telemetry items selected as of interest by the primary investigator. At this console the primary investigators uses a set of spread sheets which generate update scripts of the Sample Processing and Schedule Parameters. These update scripts are then sent to the payload operator for preparation for upload to the payload. The Experiment Display and SCL Command Consoles receive updates for their local telemetry databases across the AppleTalk network. This data is

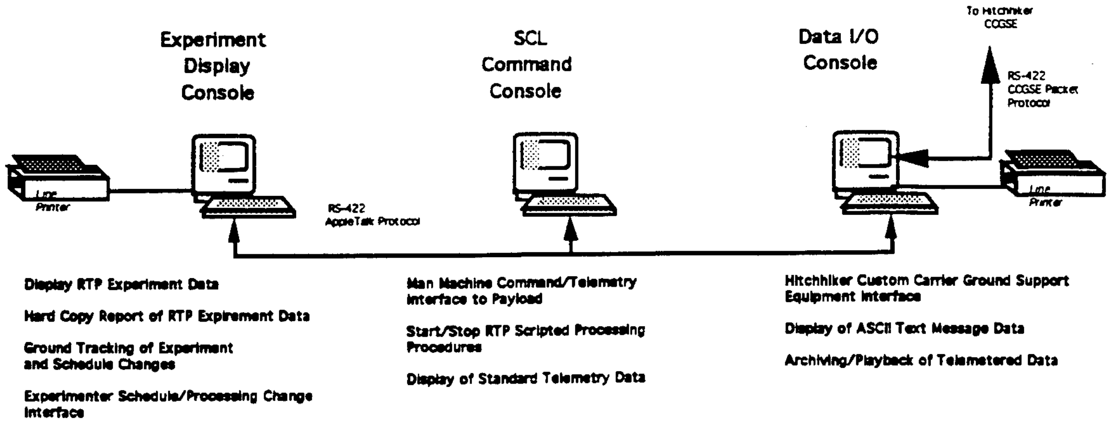


Figure 5 Requirement Mapping of Ground Station Subsystems

put on the AppleTalk network by the Data I/O Console. The SCL Command Console is used by the Payload Operator to command the payload. Commands are processed by the SCL Command Console's compiler and transmitted to the Data I/O Console for upload to the payload. At this console incoming standard telemetry data is displayed using the LabView data presentation tool. At the Data I/O Console incoming telemetry from the payload is archived, and changed telemetry items are broadcast on the network for the other consoles. The Data I/O Console is also responsible for archiving incoming telemetry data, displaying text messages from the payload, and uploading the compiled command data sent from the attached SCL Command Console.

Conclusion

The Experiment Control System for the ROMPS payload has entered the Integration and Testing phase of development. This Experiment Control System will have delivered over 30,000 lines of source code, 3 space hardened computer systems, and the electrical harness to connect them 10 months after the ROMPS Critical Design Review. The staffing level of the project was 2 full time software engineers, 2 software contractors, 1 full time electrical engineer, 1 part time assembly technicians, and 3 co-op students. Using a Randal L. Cohen's Spacecraft Characteristics Table[3] (omitting transport characteristics) the ROMPS payload is categorized as some where between complex and standard, having the characteristics displayed in Table 1. The total cost of the ROMPS project is what would normally be spent on a simple to standard payload. The financial benefits of using terrestrial based Laboratory Automation products, a commercial payload controller kernel, and standard commercial grade computers and electronic components has resulted in the development of a system with standard/complex performance at the price of a simple/standard system. The benefits to the microgravity researcher is more access to his experiment with a faster data turnaround time than he has previously been provided with.

Characteristic	ROMPS Payload	Complexity Rating
Spacecraft Telemetry Points	150+	Standard
Number of Payload Instruments	3	Standard
Payload Instrument Functionality	Several Functions; Operator Interactive	Complex
Data Retrieval	Operator Interactive	Complex
Fault Detection on the Spacecraft	Limited On-Board Safing Logic	Complex
Thermal	Some Active Control	Standard
On-Board Spacecraft Computer	Yes	Standard
Telemetry and Commanding	Real-Time Interactive Commanding and Telemetry Downlink	Complex

Table 1. ROMPS Spacecraft Complexity Data

References

1 Eric D. Cole, Tim Anderson, "Robot Operated Material Processing System ROMPS Preliminary Design Review", April 1992
2 L. Jane Hansen, Craig H. Pollock, "Space Mission Analysis", Mirocosm, Inc., Torrance, CA., Kluwer Academic Publishers, Dordrecht / Boston / London, Chapter 16, p609
3 L. David Negron, Jr., Arthur Chomas, "Space Mission Analysis", Mirocosm, Inc., Torrance, CA., Kluwer Academic Publishers, Dordrecht / Boston / London, Chapter 14, p556

Trademark Notices

A DISTRIBUTED ARCHITECTURE FOR ON-ORBIT LABORATORY AUTOMATION AND ROBOTICS USING COTS COMPONENTS

Paul Olsztyn, Michael Dobbs, and David Conrad
Environmental Research Institute of Michigan
P.O. Box 134001, Ann Arbor, Michigan 48113-4001
(313)994-1200

Abstract

The Robot-Operated Material Processing System (ROMPS) was designed to make available to the microgravity research community the same tools and mode of automated experimentation that their ground-based counterparts have enjoyed for the last two decades. This design goal was accomplished by combining commercial automation tools familiar to the experimenter community with system control components that interface with the on-orbit platform in a distributed architecture. This architecture insulates the experimenter from the details of the on-orbit platform while providing the payload engineer with the tools necessary for managing a payload. By using commercial software and hardware components whenever possible, development costs were greatly reduced when compared to traditional space development projects. Using commercial components also improved the usability of the system by providing familiar user interfaces, providing a wealth of readily available documentation, and reducing the need for training on system-specific details. The modularity of the distributed architecture implemented for ROMPS makes it very amenable for modification to other on-orbit experiments requiring robotics-based automation.

INTRODUCTION

The Environmental Research Institute of Michigan (ERIM) is a nonprofit institute specializing in the research, development, and transfer of advanced sensing, image processing, and automation-related technologies. The Space Automation and Robotics Center (SpARC) and the Space Engineering and Material Sciences (SE&MS) Department have the mission to apply ERIM's technology base in partnership programs that establish U.S. private-sector leadership in space-related commerce. SpARC approaches this mission with research and development activities that have near-term terrestrial significance and are extensible to emerging commercial opportunities afforded by the space environment.

SpARC includes in its program objectives the cultivation of a space automation supplier industry. This industry is defined to include suppliers of laboratory automation equipment such as sample preparation and manipulation, sensors, process control, data acquisition and control, and data analysis systems. Technological and economic elements of these industries are well established, with suppliers providing automation-related products and services to terrestrial "customer" industries including electronic, pharmaceutical, chemical, and other consumer product areas (i.e., food and beverage). Many of these same industries, particularly those in the electronic and pharmaceutical areas, also have an active research and development (R&D) interest in the unique properties of the space environment (microgravity, ultrahigh vacuum, view of earth, etc.).

Recognizing the commercial potential of space-based experimentation and discovery as well as manufacturing, SpARC is involved in several joint project activities with both public and private partners. While these projects focus on different niches, they all have a common goal to reduce the cost of space-based research and manufacturing through the development and deployment of technology that extends established terrestrial laboratory automation techniques to the space environment.

This paper examines the system architecture of the ROMPS Experiment Control System, which was built around two commercial off-the-shelf (COTS) technologies, Zymark's Zymate System V Controller and Interface and Control Systems' Spacecraft Command Language.

ROMPS MISSION OVERVIEW

In 1991 ERIM's SE&MS department was requested by NASA's Goddard Space Flight Center (GSFC) to propose an alternative Experiment Control System (ECS) design. The ROMPS project is a Space Shuttle Hitchhiker mission centered around the rapid thermal processing (RTP) of semiconductor materials. The ROMPS mission's short-term objective is to develop commercially promising in-space processes by using a robot-based automation

system to lower the cost of the material procedures. Figure 1 shows the payload configuration of the ROMPS Hitchhiker payload.

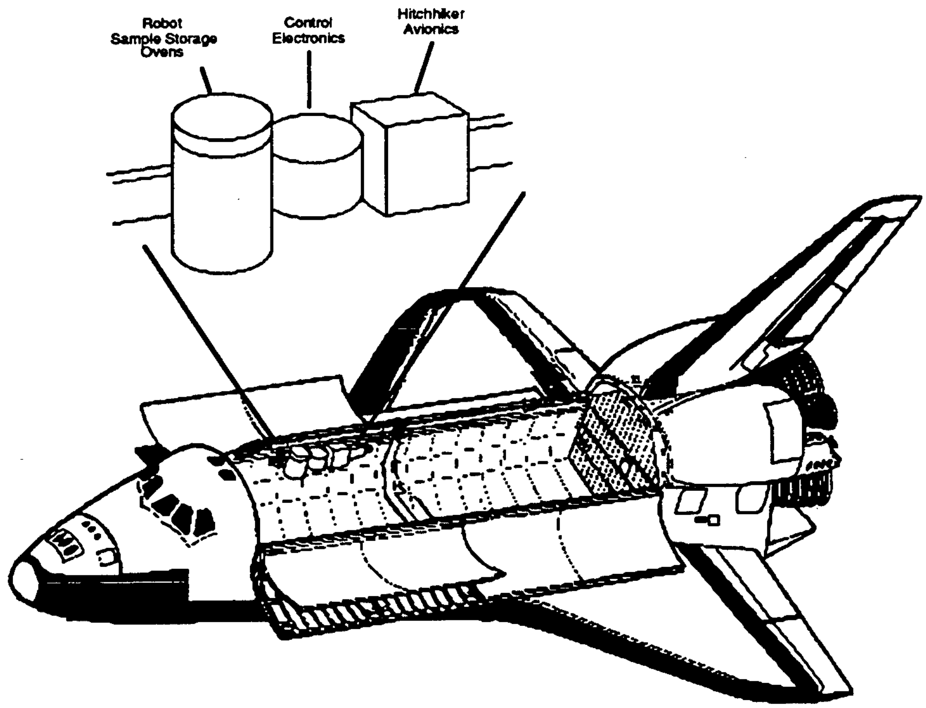


FIGURE 1. ROMPS Hitchhiker Payload Configuration.

ROMPS AUTOMATED RAPID THERMAL PROCESSING

Rapid thermal processing is a widely used industrial material processing procedure for two-dimensional semiconductor materials. In this process, a heat source capable of producing uniform surface area temperatures (usually a quartz halogen lamp) is used to melt semiconductor materials. The semiconductor materials are then allowed to cool and recrystallize. Semiconductor materials recrystallized in microgravity have larger and more uniform crystals that have better electrical properties.

ROMPS provides a robotic-based system capable of automating the RTP with up to 155 samples. In this system, a custom-built halogen lamp furnace provides the heat source for the RTP of the samples. A robot designed by GSFC moves the semiconductor samples between their storage racks and the furnace. SpARC's role in the development of ROMPS was to provide the computer, computer software, and electrical subsystems for automatically controlling and monitoring the RTP of the semiconductor materials.

ROMPS MISSION REQUIREMENTS PARTITIONING

The first step in our design process of the ROMPS Experiment Control System was to collect the experiment control system requirements and perform functional partitioning. Functional partitioning is a structured design methodology that allows the grouping of like functions in subsystem definitions without unnecessary influence from traditional subsystem boundaries (Hansen and Pollock 1992). A summary of this functional partitioning is shown below.

Robot Requirements

- Nominal motion control of 4-degrees-of-freedom robot arm/gripper
- Robot protection for end-of-travel and over-force conditions
- Calibration of robot position
- Stored robot movement sequences for rack, furnace, and station lock access
- Maintenance/reporting of robot status/sensor data

Furnace Controller Requirements

- Setting of furnace controller target power/temperature set point
- Timed on/off control of furnace halogen lamp
- Execution of a multistep temperature/time profile

Payload Controller Requirements

- Experiment/engineering data collection and transmission
- Support of Hitchhiker asynchronous serial uplink/downlink
- Scripted control of RTP processing steps
- Monitoring of experiment and engineering data with limited safety response
- Storage of initial RTP processing parameters and processing schedules
- In-flight modification of RTP processing parameters and processing schedules

Ground Station Requirements

- Human-machine command/telemetry interface to payload via Hitchhiker customer carrier ground support equipment
- HH bilevel command interface
- Start/stop RTP-scripted processing procedures
- Display/monitor telemetered engineering and experimental data
- Archiving/playback of telemetered data
- Upload/ground tracking of experiment parameter and schedule changes

TRADE STUDY OF EXISTING AUTOMATION AND ROBOTIC SYSTEMS

After completing functional partitioning, a trade study of existing data acquisition, control, robotics, and laboratory automation systems was conducted. This survey indicated that no single existing system could satisfy all of the ROMPS mission requirements, but a combination of two or more commercial systems could be modified to meet them.

Zymark's Zymate System V Controller was chosen for handling the robot and furnace controller requirement partitions. This selection was based on the superiority of Zymark's robot control software, the high modularity of the System V Controller code, and Zymark's preeminent position in the field of laboratory automation.

For the payload controller and ground station requirements, the Spacecraft Command Language (SCL) developed by Interface and Control Systems (ICS) was selected. SCL provides a payload controller kernel that combines features of a fifth-generation programming language, a rule-driven expert system, and a distributed database into a package well suited for the development of payload and ground station systems. In addition to the payload controller kernel, SCL provides a Macintosh development environment used to develop the scripts and rules executed by the SCL kernel. We decided to use the development environment as the base for a ground station command and telemetry console.

Another factor that contributed to the selection of the ICS and Zymark products as the base for our system was these organizations' willingness to work with ERIM not only as a subcontractor and product provider, but as a partner in the development process. Without this working relationship, the ability to use COTS software and hardware would be reduced to "what you see is what you get."

ADDED BENEFITS OF USING COTS FOR COMPUTER RESOURCE ESTIMATION

After functional partitioning is complete, it is standard development practice to estimate the computer resources required to meet the mission requirements. At this stage, the needed processing tasks, data requirements, and software size and throughput requirements are estimated (Hansen and Pollock 1992). This estimation process usually involves using historical size/throughput estimates of similar function points to determine the needed computer resources of the system under development. This estimation process works well until there is some function point for which there is little or no applicable historical data. This difficulty is greatly ameliorated when using COTS software and hardware products. Using the commercial SCL and EasyLab systems, we were able

to build quick prototypes to test function points for which we had no data or for which we felt there was a high degree of technical risk. This ability was particularly helpful when trying to quantify the performance characteristics of the servo control subsystem responsible for actuating the ROMPS 4-degrees-of-freedom robot arm and gripper assembly.

ROMPS DISTRIBUTED SYSTEM ARCHITECTURE

After the selection of the ICS and Zymark products as the core of the Experiment Control System, we began the process of laying out a hardware/software architecture and mapping the system requirements to specific subsystems. In Figure 2, we see the migration path of the EasyLab System V Controller, the Zymate XP Servo Controller, and the ICS SCL Macintosh development system from their native hardware architectures to a space-hardened architecture that could meet the vibrational, thermal, and power requirements specified by the Shuttle Hitchhiker Customer Accommodation Requirements. This new architecture also provides interfaces for the engineering and experimental data collection sensors and discrete command interfaces, which are not needed in a terrestrial laboratory system.

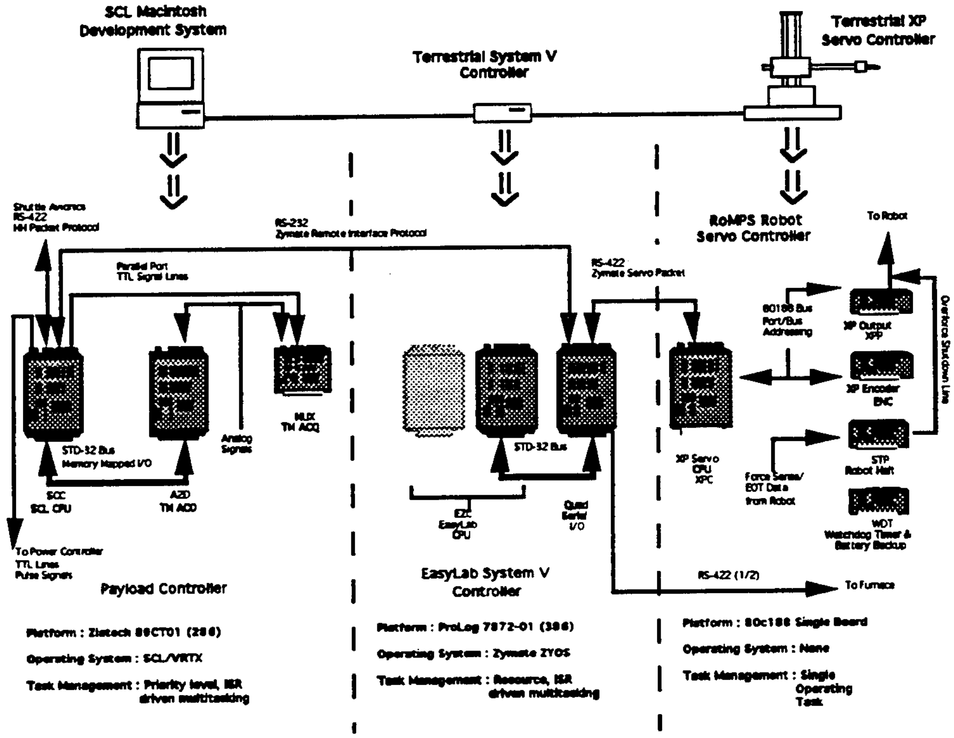


FIGURE 2. Migration of Terrestrial Architecture to a Space-Hardened Architecture.

USE OF COTS SINGLE-BOARD COMPUTERS AND PERIPHERALS

First and foremost to meet our commercial customer objectives, the use of COTS components reduces the cost of experimental systems. Second, COTS components accelerate the space investigator's access to familiar or state-of-the-art technology.

Reduced Cost

While it is difficult to compare the cost of conventional space "qualified" systems with space "ruggedized" systems, the following comparisons are of interest:

- An accepted estimate of developing full space qualified avionics systems (with the same mass as the ECS) ranges from $1.1M to 7.6M (Wong 1992).

- Our approach, maximizing COTS content and accepting system reliability estimates of 97 percent versus 99.99 percent, reduced the actual cost (including mission support) to $750K.

Acceleration of Technology Introduction

As shown in Table 1, the introduction of computers to space missions has significantly lagged behind the same technology's commercial introduction. While a significant element of the lag is the lengthy and costly delay associated with radiation hardening, there are many missions that do not require this characteristic. One of our goals is to accelerate the introduction of a new computing capability where the need for high performance at low cost outweighs susceptibility to radiation.

Similar lags exist in the amount of computing memory. In the 1980s, 256 Kbytes was considered the limit; today, we routinely provide 2 Mbyte or more, *per processor*. A significant related development in 1994 was the introduction of the "solid-state recorder" and a space-qualified disk drive as replacements for the venerable tape recorder.

TABLE 1. Technology Introduction Acceleration.

Computer Technology	Commercial Introduction	Space Mission Launch	Delay in Years
x186/188 also PC-XT	1983	1989	6
x286 also PC-AT	1984	1989 est	5
x386	1988	1993 est	5
x486	1989	1995 est	6
Pentium	1993	1996 est	3
PowerPC	1994	1996 est	2

Compound Savings

When using COTS components, there are compounded savings. In addition to the direct reduction in the cost of the flight experiment, there are significant savings in related areas:

- The overall project risk is reduced because there are fewer new items to develop, and critical items are available earlier in the project cycle.
- The ground and data processing costs are reduced because the COTS flight system is usually compatible with COTS (i.e., low-cost) ground systems.
- Life-cycle costs are reduced because development tools, documentation, system upgrades, and so forth, are developed for the commercial market and are thus *free*.

Rapidly Increasing COTS Content

In less than three years, we have gone from systems with no COTS content to systems with better than 60 percent COTS content (see Table 2). In 1992 we achieved 50 percent COTS contents; by 1996 we are planning systems with 80 percent COTS content. We have carried this philosophy back into other areas of our parent organization (ERIM) as one means to reduce the increasing cost of R&D and prototype development. With each new customer and program, we continue to reduce the custom content of our systems while we improve performance.

TABLE 2. COTS Content Increase.

	ARD 1992	ROMPS 1994	AWCS 1996
Custom modules	4	4	2
COTS modules	4	6	9
% COTS content	50 %	60 %	81%

Lessons Learned

While the use of COTS systems can achieve impressive cost savings and performance gains, there are several traps that need to be avoided. It is important that the development manager budget sufficient calendar time to cover the

learning curve associated with the products and tools. It is equally important to budget adequate staff training in the use of these products and tools. These minor investments will provide more than ample return to the program and those programs that follow. Equally important, is that the primary investigator (PI), customer, or end-user not require that every feature listed in the product's manual be used. With any product, not all features are created equally, and some will simply never work.

Interfacing the Payload Controller to the EasyLab System V Controller

One of the first technical hurdles in establishing the ROMPS distributed architecture was in creating a simple interface between the payload controller's Spacecraft Command Language kernel and the EasyLab System V Controller interpreter. In a terrestrial lab, the EasyLab System V Controller uses a high-speed serial interface to a PC to provide a user interface. The communication throughput requirements necessary to implement this interface could not be supported by the Hitchhiker's 1200-baud serial downlink. Discussions with Zymark personnel revealed that the System V Controller contained a remote control interface designed to allow a foreign computer to control and communicate with the System V Controller's command interpreter. Consequently, the Spacecraft Command Language was modified to include two new directives, Zymate_Command and Zymate_Query. These two new directives allow EasyLab commands and data requests to be issued to the on-orbit System V Controller via the payload controller. This pass-through mechanism allows experimenters to communicate directly with the experiment devices, using a command interface well known and accepted in the user community.

THE FLIGHT SEGMENT ARCHITECTURE

Figure 3 summarizes the mapping of functional requirements onto the flight subsystems of the Experimental Control System. This figure also reveals the hierarchical levels of experiment control performed by the ECS subsystems. Moving from the SCL Experiment Supervisor on the far left to the ROMPS Robot Servo Controller on the far right, the command interfaces become less abstract and more device-specific.

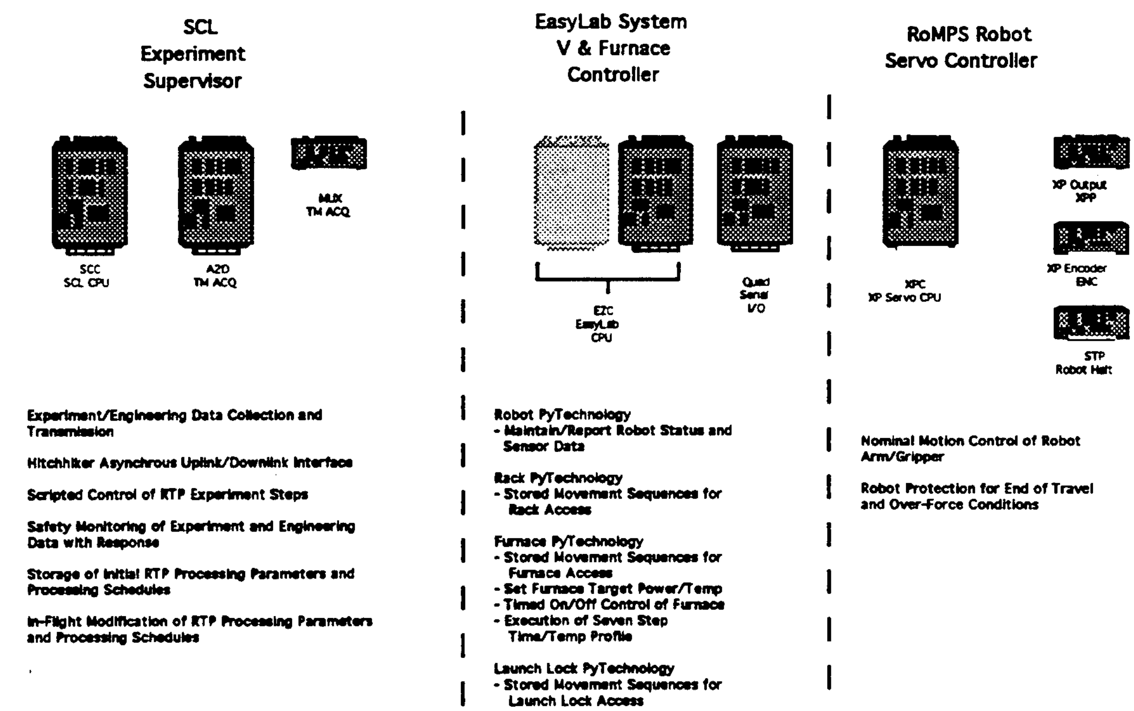


FIGURE 3. Requirements Mapping of Flight Subsystems.

At the SCL Experiment Supervisor, the control of the experiment is at a very high abstraction level. This subsystem executes the RTP processing scripts, which set sample processing parameters and initiate laboratory unit operations by sending commands to the Zymate System V Controller. This subsystem also monitors the health and safety of the payload, and halts the RTP script if it detects an anomalous condition, such as temperature of the furnace structure exceeding the safe operating limit.

The Zymate System V Controller executes the steps of the laboratory unit operations initiated by the SCL Experiment Supervisor. Device-specific commands are sent to the ROMPS Robot Servo and Furnace Controllers to complete these laboratory unit operations. At the ROMPS Robot Servo Controller, the device-specific robot commands sent from the EasyLab System V Controller are processed. This subsystem also protects the robot from harming itself during movement command sequences.

Many of the disadvantages of a distributed architecture are eliminated in this system by the use of simple, robust, and well-tested communication protocols.

THE GROUND SEGMENT ARCHITECTURE

Figure 4 summarizes the mapping of functional requirements onto the ground station subsystems of the Experiment Control System. Again, the hierarchical levels of command and control are shown. Decreasing levels of experiment control abstraction are shown as one proceeds from left to right across this figure. At the far left, the experiment display console displays and prints the telemetry items that have been selected by the primary investigator. At this console, the primary investigators use a set of spreadsheets to generate update scripts to be executed against the on-board sample processing and schedule parameters. These update scripts are then sent to the payload operator for preparation for upload to the payload. The experiment display and the SCL command consoles receive updates for their local telemetry databases across the AppleTalk network. These data are placed on the AppleTalk network by the data input/output (I/O) console.

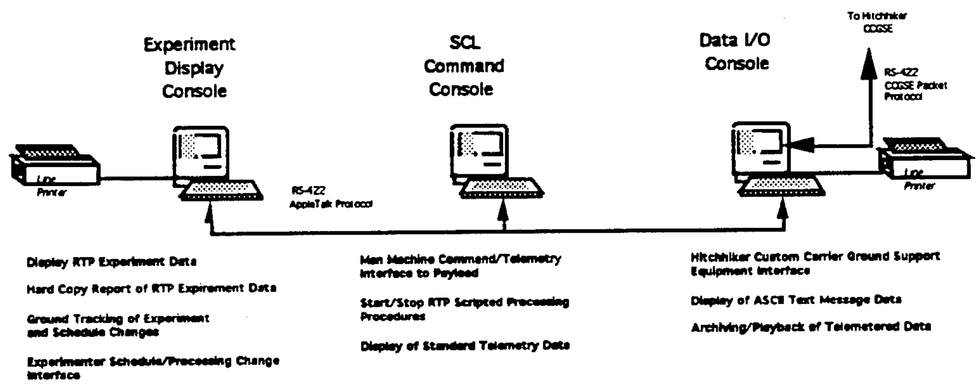


FIGURE 4. Requirement Mapping of Ground Station Subsystems.

The SCL command console is used by the payload operator to command the payload. Commands are processed by the SCL command console's compiler and transmitted to the data I/O console for upload to the payload. At this console, incoming standard telemetry data are displayed using the LabView data presentation tool. At the data I/O console, incoming telemetry from the payload is archived, and changed telemetry items are broadcast on the network for the other consoles. The data I/O console is also responsible for archiving incoming telemetry data, displaying text messages from the payload, and uploading the compiled command data sent from the attached SCL command console.

All of the application programs that make up the ROMPS ground station use the standard Macintosh window/icon/pointer/mouse user interface paradigm. This allows users to concentrate on learning the functionality of the applications without worrying about the details of the human-machine interface.

MISSION OPERATIONS FOR MATERIAL PROCESSING PROCEDURES

One of the driving philosophies of the ROMPS system architecture was to enable the PIs to interact with the payload in a meaningful way without the need of a payload operator to translate every command request into a set of payload directives. To meet this requirement, it was necessary to identify the operations the PIs would be interested in initiating and the telemetry data that would be of immediate value to them during mission operations. After several conferences with the ROMPS primary investigators, it became clear that the PIs would like to specify what samples to run during a given operational period, and what power and at which times the samples in that run would be processed. To accommodate this method of command and control, a set of on-board SCL mission scripts was implemented that would take as an input a list of samples to be processed. These scripts would then

access a global table containing the processing parameters for all the samples aboard the payload in order to determine the location of the sample and the power and time settings for the samples contained in the inputted schedule list.

To enable the PIs to interact with the ROMPS payload, it was necessary to provide them with a data-entry mechanism for specifying a list of samples to be processed, and the processing parameters for these samples. We decided that because these data were essentially tabular in nature, a spreadsheet could be used as the data input mechanism. Custom spreadsheet macros would be created to allow the user to generate lists of samples to be processed and processing parameter specifications.

Microsoft Excel was used to create two data input spreadsheets, one for generating schedules, and one for specifying processing parameters. Figure 5 shows the Parameter Table spreadsheet with processing data for the ten samples entered. This table allows the PI to specify at what power and for how long samples are to be processed. Four macro buttons at the top are used to check the validity of the data entered, and to create SCL scripts that map the data in the spreadsheet cells to the corresponding entry in the on-orbit parameter table. Figure 6 shows the Schedule Table spreadsheet with a processing run of five samples entered. The Total Processing Time, Sample Set ID, Slot ID, and SEQ# columns are filled in with data from the Parameter Table Spread sheet when an entry is made in the Sample No. column. The Sample Processing Time column totals all the time fields for a specified sample, and then adds the robot transport time to this value.

Table Patches = 0

Export Entire Table | Check Entire Table | Record Table Updates | Export Table Updates

Sample Set ID	Slot ID	SEQ #	Sample No.	Rack No.	Slot No.	Power 1 Watts	Time 1 in sec.	Power 2 Watts	Time 2 in sec.	Power 3 Watts	Time 3 in sec.	Power 4 Watts	Time 4 in sec.	Power 5 Watts	Time 5 in sec.	Power 6 Watts	Time 6 in sec.	Power 7 Watts	Time 7 in sec. Grav Sen	Cool Time
1-Cole/Si	E27	C1	001	5	26	125	20	0	0	0	0	0	0	0	0	0	0	0	1	0
1-Cole/Si	E28	C2	002	5	27	25	15	125	20	0	0	0	0	0	0	0	0	0	1	0
1-Cole/Si	F12	S1	003	6	08	140	30	0	0	0	0	0	0	0	0	0	0	0	1	0
1-Cole/Si	F13	S2	004	6	09	140	30	0	0	0	0	0	0	0	0	0	0	0	1	0
1-Cole/Si	F14	S3	005	6	10	140	60	0	0	0	0	0	0	0	0	0	0	0	1	0
1-Cole/Si	F15	S4	006	6	11	140	60	0	0	0	0	0	0	0	0	0	0	0	1	0
1-Cole/Si	F16	S5	007	6	12	140	120	0	0	0	0	0	0	0	0	0	0	0	1	0
1-Cole/Si	F17	S6	008	6	13	140	120	0	0	0	0	0	0	0	0	0	0	0	1	0
1-Cole/Si	F18	S7	009	6	14	142	30	0	0	0	0	0	0	0	0	0	0	0	1	0
1-Cole/Si	F19	S8	010	6	15	142	30	0	0	0	0	0	0	0	0	0	0	0	1	0
1-Cole/Si	F20	S9	011	6	16	142	60	0	0	0	0	0	0	0	0	0	0	0	1	0
1-Cole/Si	F21	S10	012	6	17	142	60	0	0	0	0	0	0	0	0	0	0	0	1	0

FIGURE 5. Parameter Table Data Entry Spreadsheet.

Schedule ID = 12

Record Schedule | Export Schedule

Sample No.	Sample Processing Time	Total Processing Time	Sample Set ID	Slot ID	SEQ #
		00:00:00			
2	00:00:36	00:03:36	1-Cole/Si	E28	C2
4	00:00:31	00:07:07	1-Cole/Si	F13	S2
7	00:02:01	00:12:08	1-Cole/Si	F16	S5
9	00:00:31	00:15:39	1-Cole/Si	F18	S7
10	00:00:31	00:19:10	1-Cole/Si	F19	S8

FIGURE 6. Schedule Table Data Entry Spreadsheet.

After a parameter table patch or batch schedule SCL script has been created, it is given to the payload operator who compiles the script into the ground-based version of the flight project. Parameter table patches and batch schedules are then uploaded to the payload. After a parameter table patch has been uploaded, it must be executed by the payload operator in order for the global arrays used to store the parameter table to be updated.

Figures 7 and 8 summarize the process of up-linking parameter or schedule changes to the flight system. In each case, the appropriate spread sheet is used to create a text file containing an SCL script that, when executed, will update the global data structures used to store the sample processing parameters or the sample schedule list. After the text file is created, it is placed into an AppleShare folder where it can be accessed by the payload engineer. The payload engineer compiles the script and then uploads it to the on-orbit Experiment Control System.

POST-MISSION DATA ANALYSIS

During the mission, data are archived by the DataIO application running on the data I/O console into playback files. These playback files archive the incoming telemetry packets and the outgoing command packets. The DataIO application can replay these playback files and will drive all the attached processes as if the data were coming in real time. This allows the LabView application running on the Experiment Display Console to display archived data in order to review mission events. When playing back archive file, DataIO can preserve the relative time domain in which the telemetry occurred or at the maximum speed that the system is capable of performing.

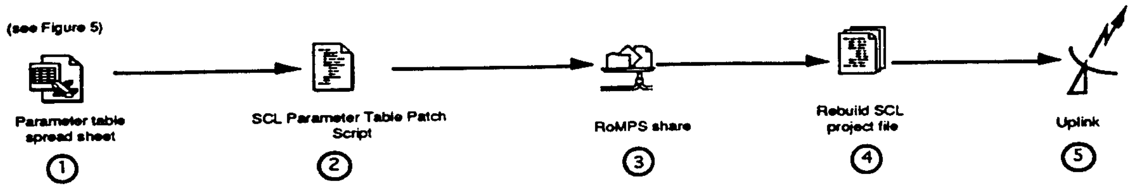
In addition to the ability to drive attached telemetry display applications, the DataIO application can produce spreadsheet-importable text files containing selected telemetry items. DataIO performs the necessary data translations so the telemetry items are presented in user units. Figure 9 shows the output of such a playback file. Notice that the data acquisition time and ground station packet receipt time are automatically generated for each telemetry point recorded in the file. In this figure, the columns marked DB REC 128 and DB REC 129 are the two telemetry points selected for extraction before the archive file was played back. DB REC 128 is a temperature sensor and DB REC 129 is a counter used by the onboard telemetry output task to stamp each outgoing packet.

CONCLUSION

The Experiment Control System for the ROMPS payload has completed integration and testing and is now awaiting launch aboard the STS-64.

In just 13 months after the ROMPS Critical Design Review, the Experiment Control System delivered over 30,000 lines of source code, three space-hardened computer systems, and the electrical harnessing to connect them. The staffing of this project included two full-time software engineers, two software contractors, one full-time electrical engineer, one part-time mechanical engineer, one part-time assembly technician, and three co-op students.

Using Randal L. Cohen's Spacecraft Characteristics Table (Negron and Chomas 1992) (omitting transport characteristics), the ROMPS payload is categorized as falling between "complex and standard," having the characteristics displayed in Table 3. The total cost of the ROMPS project is what would normally be spent on a "simple to standard" payload. The financial benefits of using terrestrial-based laboratory automation products, a commercial payload controller kernel, and standard commercial-grade computers and electronic components have resulted in the development of a system with high performance at a low price.



1 At the SCL Script Development Console, the PI invokes the Excel "APC parameter table" spread sheet, clicks the "Record Table Updates" button, and changes the desired processing parameters by entering new values into the appropriate cell.

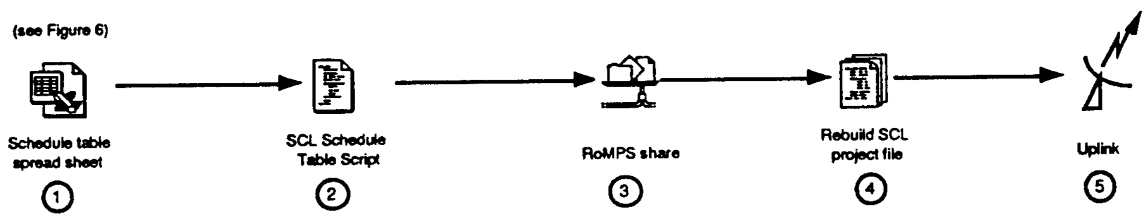
2 After the desired processing parameters have been modified, the PI checks the validity of the data entered by clicking on the "Check Entire Table" button. This button will check the range of the annealing time and temperature parameters. The PI then exports the processing parameter data by clicking on the "Export Table Updates" button. This creates the parameter table "patch" script to update the SCL global arrays storing the process parameter table.

3 The SCL Parameter Table Patch is then transferred to the RoMPS Share folder where the Payload Engineer can access it from the SCL Command Console.

4 The SCL Parameter Table Patch script is then compiled by the Payload Engineer into the SCL Project File.

5 The compiled Parameter Table PatchScript is now ready for uplink to the RoMPS Experiment Supervisor where it can be executed to modify the on-orbit Processing Parameter Table

FIGURE 7. Creating Parameter Table Patches.



1 At the SCL Script Development Console, the PI invokes the Excel "APC schedule table" spread sheet, clicks on the "Record Schedule" button, and creates a schedule by typing in the sample identifiers for the processing run. The spread sheet will compute the time to process that sample and maintain running total for all samples in the schedule.

2 After the desired schedule has been created, the PI exports the scheduling data by clicking on the "Export Schedule" button. This creates the sSchedule Table Script which loads the SCL global array specifying the samples to be processed.

3 The Schedule Table Script is then transferred to the RoMPS Share folder where the Payload Engineer can access it from the SCL Command Console.

4 The Schedule script is then compiled by the Payload Engineer into the SCL Project File.

5 The compiled Schedule Table Script is now ready for upllink to the ROMPS Experiment Supervisor where it can be executed by the on-orbit Automated Processing Cycle script.

FIGURE 8. Creating New Schedule Tables.

GROUND TIME-STAMP	FLIGHT MET	DB REC 127	DB REC 128
Wed Jul 20 18:03:36 1994	0: 2:21: 1	22	119
Wed Jul 20 18:03:36 1994	0: 2:21: 2	22	120
Wed Jul 20 18:03:36 1994	0: 2:21: 3	22	121
Wed Jul 20 18:03:36 1994	0: 2:21: 4	22	122
Wed Jul 20 18:03:37 1994	0: 2:21: 4	22	123
Wed Jul 20 18:03:37 1994	0: 2:21: 5	23	124
Wed Jul 20 18:03:37 1994	0: 2:21: 6	22	125
Wed Jul 20 18:03:37 1994	0: 2:21: 7	23	126
Wed Jul 20 18:03:37 1994	0: 2:21: 8	23	127
Wed Jul 20 18:03:37 1994	0: 2:21: 9	23	128
Wed Jul 20 18:03:37 1994	0: 2:21:10	23	129
Wed Jul 20 18:03:37 1994	0: 2:21:11	23	130
Wed Jul 20 18:03:37 1994	0: 2:21:12	24	131
Wed Jul 20 18:03:37 1994	0: 2:21:13	23	132
Wed Jul 20 18:03:38 1994	0: 2:21:14	23	133
Wed Jul 20 18:03:39 1994	0: 2:21:15	23	134
Wed Jul 20 18:03:40 1994	0: 2:21:16	23	135
Wed Jul 20 18:03:41 1994	0: 2:21:17	25	136
Wed Jul 20 18:03:42 1994	0: 2:21:18	22	137
Wed Jul 20 18:03:43 1994	0: 2:21:19	22	138
Wed Jul 20 18:03:44 1994	0: 2:21:20	22	139
Wed Jul 20 18:03:44 1994	0: 2:21:21	22	140
Wed Jul 20 18:03:45 1994	0: 2:21:22	22	141
Wed Jul 20 18:03:46 1994	0: 2:21:23	22	142

FIGURE 9. Spread Sheet Formatted Telemetry Data Extracted by DataIO.

TABLE 3. ROMPS Spacecraft Complexity Data.

Characteristic	ROMPS Payload	Complexity Rating
Spacecraft Telemetry Points	150+	Standard
Number of Payload Instruments	3	Standard
Payload Instrument Functionality	Several Functions; Operator Interactive	Complex
Data Retrieval	Operator Interactive	Complex
Fault Detection on the Spacecraft	Limited On-Board Safing Logic	Complex
Thermal	Some Active Control	Standard
On-Board Spacecraft Computer	Yes	Standard
Telemetry and Commanding	Real-Time Interactive Commanding and Telemetry Downlink	Complex

NEXT MISSION

ERIM's ongoing challenge is to adapt the ECS to automate the molecular beam epitaxy (MBE) processing methods to be performed during the upcoming third flight of the Wake Shield Facility. This mission will extend the capabilities of the ECS with the addition of process control and process sensors. Keeping with the COTS philosophy, and modular architecture, this set of new capabilities will include preintegrated add-on modules that perform real-time machine vision and indirect temperature sensing for the purpose of in-line process control.

Microgravity researchers will benefit through increased access to their experiment and a faster data turnaround time.

References

Hansen, L.J. and C.H. Pollock (1992), *Space Mission Analysis and Design*, 2nd edition, Kluwer Academic Publishers, Boston, Massachusetts:609-616.

Wong, R. (1992), *Space Mission Analysis and Design*, 2nd edition, Kluwer Academic Publishers, Boston, Massachusetts:666.

Negron, L.D., Jr. and A. Chomas (1992) *Space Mission Analysis and Design*, 2nd edition, Kluwer Academic Publishers, Boston, Massachusetts:556.

Acknowledgments

ERIM would like to acknowledge the NASA Goddard Space Flight Center personnel who funded this effort under NAGW-1198, NAG5-1517, and NAS5-32471. We deeply appreciated this research and development opportunity, and their willingness and support throughout this unorthodox and challenging effort. Similarly, this program could not have been accomplished without the tireless technical and financial support of the two commercial product vendors—the team at Interface and Control Systems for accomplishing many modifications and improvements to SCL, and the Zymark Corporation for their support in modifying the Easylab and the XP servo software. Our thanks to the many unnamed individuals who contributed invaluable effort and advice when it was most needed.

Trademark Notices

AMPS

Advanced Modular Power Systems **Space Resources Division**

Experiment Control System™ II

ECS II™ BACKGROUND

- Low cost, adaptable, experiment control system
  - Commercial off-the-shelf hardware/software approach
  - Event-driven or scripted control, data collection, process monitoring
  - Interface to Hitchhiker asynchronous serial protocol
  - Interfaces to COMET (XMODEM) and Wake Shield Facility available
- Developed under NASA 'Robot Operated Materials Processing System' program
  - Flown on STS-64 mission
  - 100% successful process control in automatic mode
  - Error free through South Atlantic Anomaly
- Ground station control
  - Graphical display of status
  - Archiving/playback of telemetry data
  - Configuration control of modified schedules and parameters
- Selected for future missions
  - Robotics Controller for Wake Shield III on STS-78 (SVEC)
  - AMTEC In-STEP Experiment (JPL)
  - ROMPS-2 (GSFC)

ECS II™ SYSTEM ADVANTAGES

- Technically mature product - demonstrated performance in space
- Low cost - avoids expensive alternatives
- Provides real time access to experiment
  - Automatic operation
  - Reprogram on-orbit
  - Command directive or time based script execution
- Ground Station Equipment
  - MAC Centris, Quadra and PowerMac (low cost)
  - LabView™ Graphical Telemetry Displays
  - Excel™ Telemetry Analysis and Graphing
- Qualified for ELV, STS and Hitchhiker missions
- Rapid delivery ~ 3-6 months ARO
- Application development units available in (2-4 weeks).

· See back for Specifications and Options ·

Advanced Modular Power Systems, Inc.
Space Resources Division
4667 Freedom Drive
Ann Arbor, Michigan 48108

Tel: (313) 994-1200 Ext. 2476
Fax: (313) 668-8957

AMPS

Advanced Modular Power Systems **Space Resources Division**

Experiment Control System™ II

GENERAL SPECIFICATIONS

- Command and Telemetry processor
  - V53 processor running SCL™
  - 1 Mbyte RAM and 1 Mbyte ROM
  - Two RS232/442 serial channels capable of 19.2 kbaud
  - 32 configurable digital I.O. lines
- Analog to Digital Converter
  - 12 Single ended inputs, ±10 V range
  - 12 bit resolution
- Analog Multiplexer
  - Two banks of 16 differential or 32 single ended inputs
  - Each bank - selectable gain 1-1000
  - Signal conditioning/excitation for 16 YSI thermistors
  - ± 10 V input range
- Size (in) 12 x 7 x 8.5 , LxWxH (6 slot enclosure)
- Weight - 10 lbs
- Power - 6 Watts (min)
- Input supply range - 18 V to 40 V
- Operating temperature -10 °C to + 55°C (in vacuum)
- Vibration - 12 G_{rms} (per HH CARS)

OPTIONS

- Quad Serial I/O
  - Two RS232 channels
  - Two RS232/422 channels
  - Opto-isolation and FIFO on all channels
- Expanded Program Memory
  - Two Mbytes per card
  - Battery Backed option
- Thermocouple Signal Conditioner
  - 32 Channels for type K, J, T, S, R
  - Cold junction compensation
- Robotics Servo Control System
  - 8 Axis w/ EOT and HOME
  - 16-bit D/A Output
  - Incremental Encoder or 12-bit A/D
- Dedicated Data Recorder
  - 40+ MByte Capacity
- Electronic Load Controller
  - 250W Source/Sink Op-Amp
  - 16-bit Resolution

COST

Call for Quote and Lastest Options List

Advanced Modular Power Systems, Inc.
Space Resources Division
4667 Freedom Drive
Ann Arbor, Michigan 48108

Tel: (313) 994-1200 Ext. 2476
Fax: (313) 668-8957

Across The Backplane

The Official STD 32® Newsletter

Fall/ Winter 1994-95

No. 6

New 386- and 486-based Processors Expand Industrial Computer Offerings

486SLC2 Powers New Computer

A new 50 MHz 486 CPU from **VersaLogic Corporation** (Eugene, OR) combines the industrial STD 32® architecture and complete "AT" software-compatibilty. The STD 32 Bus provides 16- and 32-bit performance, backward compatibility with 8-bit STD-80 boards, a small 4.5" by 6.5" format, and reliability in industrial applications.

The VL-486-2 features the 50 MHz 486SLC2, up to 4 Mbytes of RAM, 128 to 512 Kbytes of flash/ROM, two COM ports, an LPT port, an IDE interface, a floppy disk interface, front panel DMA and interrupt connectors, a watchdog timer, Vcc sensing reset, and a 387SX math coprocessor socket. It can operate as a multiprocessor within an STD 32 system, and is supported by development systems for DOS and non-DOS systems.

In addition to a full line of industrial CPUs, VersaLogic offers analog, digital, serial and combination I/O boards plus industrial card cages and power supplies.

For more information, call Karen McConnell, VersaLogic Corporation, (800) 824-3163.

New Single Board 386 EX Computers Serve Embedded Applications

New modestly priced STD 32 computers from **Ziatech Corporation** (San Luis Obispo, CA) are the first industrial computers to use the Intel386 EX CPU, Intel's new microprocessor for embedded designs.

The ZT 8903 and ZT 8904 cost effectively incorporate the PC-compatible, embedded features of the Intel386 EX processor into the compact, rugged STD 32 format.

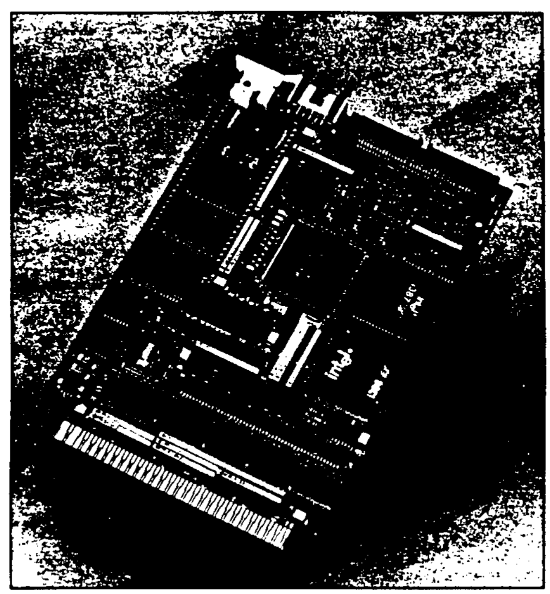
Complete Industrial PC On-Board

The PC software-compatible ZT 8903 SBC includes up to 5 Mbytes of Pseudo Static RAM, 1 Mbyte of flash memory, 24 lines of digital I/O, two serial ports, a parallel port, and a watchdog timer, with local bus video and math coprocessor options.

The ZT 8904 SBC expands this feature set, adding two RS-232/485 serial ports, 128 Kbytes of SRAM, and an optional IDE disk drive subsystem.

Multiprocessing Capability, Too

In addition to stand-alone and single processor applications, the new single board computers will operate with other processors in an STD 32 multiprocessing system such as Ziatech's STD 32 STAR SYSTEM™. The STAR SYSTEM is designed for applications requiring a combination of real-time control with a graphical user interface. It allows up to seven DOS or QNX-based processors to share peripherals and I/O in a single system.



Ziatech's new ZT 8904 Single Board 386 EX Computer

Other New Products

Ziatech has also recently introduced an STD 32 enclosure with two ISA slots, new single-slot hard and floppy disk drives, and an intelligent multichannel serial controller.

For more information, contact the Sales Department at Ziatech Corporation, (805) 541-0488.

Continues on page 3

STD 32 Space Experiments

An STD 32-based computer system developed by the Environmental Research Institute of Michigan (ERIM) helped researchers conduct experiments aboard a recent Space Shuttle flight, and is scheduled for future space missions.

Created by a team from ERIM's Space Engineering and Material Science Department, the Experiment Control System™ (ECS™), performs autonomous control of NASA's Robotic Operated Material Processing System.

Designed for Experiments

"The ECS is a cost-effective, open architecture package of STD 32 hardware and off-the-shelf software, designed for experiments on the Space Shuttle, sounding rockets, small satellites, and the Space Station," says ECS Product Manager Michael E. Dobbs.

The STD 32-based control system supports several levels of capability, according to the user's needs. These capabilities include:

- standard experiment control, telemetry data acquisition, display and archive
- expert system and rule-based automatic anomaly detection and response
- control of mechanisms, automation and robotic systems
- process control

Another ECS system has been selected as the manufacturing and process control system for the first U.S. orbiting maufacturing satellite, the Wake Shield Facility, Mission 3, planned for mid-1996.

For more information, contact Michael Dobbs at (313) 994-1200, ext. 2476.

New Products *Continued from page 1*

MIL-STD-1553 and ARINC Interfaces

Excalibur Systems, Inc. (Fresh Meadows, NY) has recently added two new products to its line of Avionics communications equipment. The EXC-1553RBU is a single and two-channel interface card that is compatible with MIL-STD 1553A and 1553B. Each channel has an on-board 32 Kbyte dual-port RAM interface, and operates on an independent basis. These cards are available in commercial, extended temperature and ruggedized versions, and are suitable for airborne applications.

MAGIC Avionics Card

Excalibur also produces the EXC-3000STD, known as the MAGICard. This card can be populated with various Avionics protocols at one time, including ARINC-429/575/568/582/561 and RS-232/422/485.

For more information, contact Excalibur Systems, Inc., (718) 357-3500 or (213) 936-8236.

DSP Motion Controller

Motion Engineering, Inc. (Santa Barbara, CA) offers a technologically advanced STD/DSP motion controller that controls both servos and steppers with one board and programming interface. Motor types can be mixed on a single board with the STD/DSP's full range of control and I/O capability.

Outputs include 16-bit resolution analog signals for brush and brushless servo motors and high-resolution step and direction signals for open-loop and closed-loop steppers.

For more information, contact Motion Engineering, (805) 962-5409.

"...a cost-effective package of STD 32 hardware and off-the-shelf software designed for experiments on the Space Shuttle, sounding rockets, small satellites, and the Space Station."

STD 32®
The Embedded Computer℠

Inside...

New Products and Embedded Applications

1995 Embedded Computing Shows to Feature STD 32®

STD 32 manufacturers will exhibit their products across the country in 1995 by participating in the *Embedded Computing Show Featuring PC Advanced Technology*, a series of table top shows sponsored by the RTC Group. The RTC Group also manages the popular Real-Time Shows.

As its new name implies, the Embedded Computing Show highlights the products and services of companies in the embedded computer industry. The one-day show will make its 1995 debut on January 24 in San Diego, and January 26 in Irvine. The show then travels to other major cities, including Boston, Princeton, Dallas, Austin, and Santa Clara, as well as special sites and military bases such as the Jet Propulsion Laboratory and Patterson Air Force Base.

For a calendar of the 1995 Embedded Computing Show dates and locations, contact Steve Grimaldi of the RTC Group at (714) 489-7575.

The STD 32 Special Interest Group
11766 Wilshire Boulevard
Los Angeles, California 90025-9639 USA

Bulk Rate
U.S. Postage
PAID
Los Angeles, CA
Permit No. 01901

MIKE DOBBS
ERIM
P O BOX 134001
ANN ARBOR MI 4811[illegible]

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

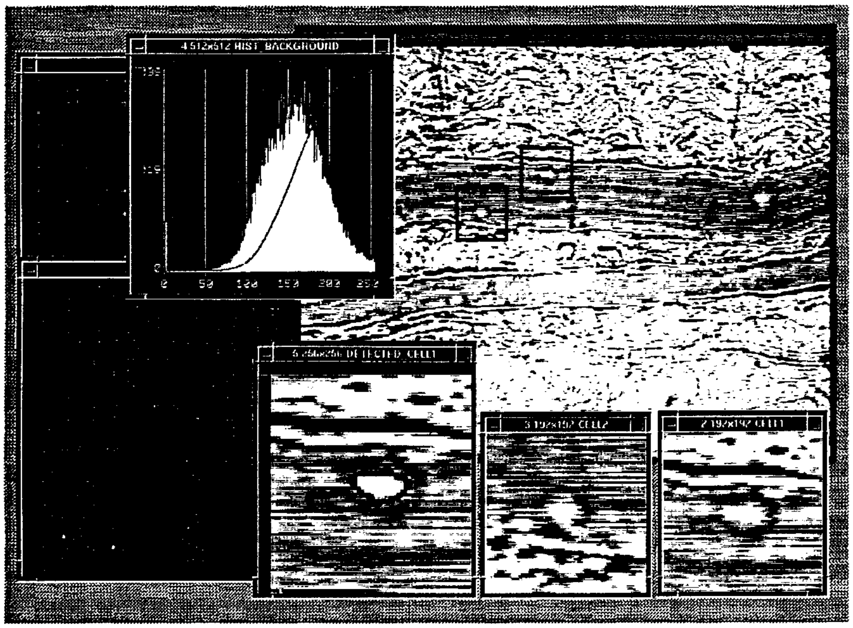
BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 71834 LOS ANGELES, CA

POSTAGE WILL BE PAID BY ADDRESSEE

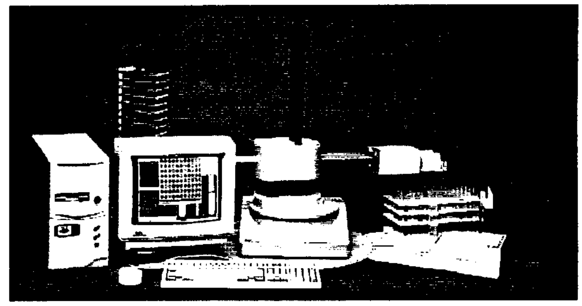
The STD 32 Special Interest Group
11766 Wilshire Boulevard
Los Angeles, California 90025-9639 USA

Vision Instruments Machine Vision Product Information

VI-10 Machine Vision Instrument



The machine vision technology that helped win 'Desert Storm' brought to your laboratory



The affordable machine vision instrument for your laboratory automation & industrial tasks

What the VI-10 is and is not:

✓ A packaged PC solution to your machine vision problems built upon our Vision Engine Software.

✓ An extensible systems which readily incorporates your existing algorithms and programs.

✓ A UNIX based package complete with the networking, display, and IPC tools needed to integrate a vision solution into your automation system.

✓ A robust image processing workhorse for production class applications which produces hard quantative and qualitative data.

Ø NOT Another software package for you to buy and struggle to learn alone.

Ø NOT A closed toolbox to which you must convert your existing efforts.

Ø NOT A stand-alone environment unable to interface with the diverse elements of an automation system.

Ø NOT Just a GUI which produces pretty pictures instead of useful answers.

Vision Instruments Inc.
PO Box 130522
Ann Arbor, Michigan
48113-0522

Tel: (313) 994-9260
Fax: (313) 994-8618
E-mail: info@vi.com

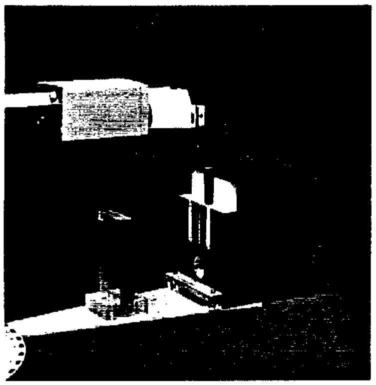
Vision Instruments

About Vision Instruments

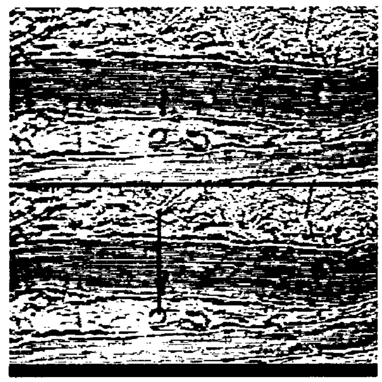
Our corporate goal is to provide practical solutions to complex imaging problems using affordable PC-based technology and our proven Vision Engine software. This software is licensed from the Environmental Research Institute of Michigan, the world's largest research organization dedicated to image processing. Originally developed for DoD tasks of detecting uncooperative targets, this software is now being used for commercial applications. By leveraging our engineering staff's decades of experience delivering image processing solutions with this technology our corporate goal is attained. We are thus both a product and service provider. This approach allows us to deliver vision enabled instruments which plug in and provide answers.

We have delivered machine vision solutions in...

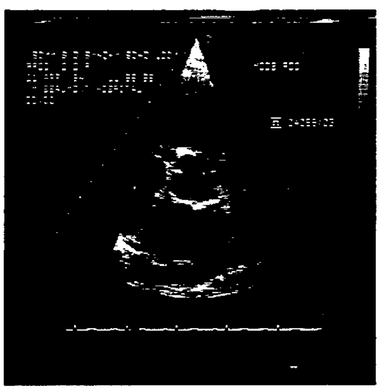
Laboratory Robotics



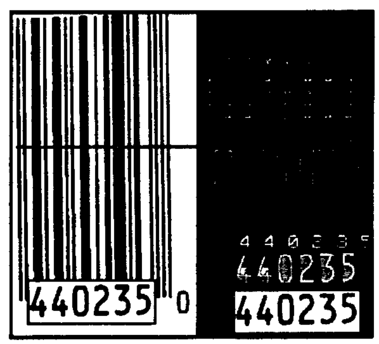
Biotechnology



Medical Imaging



Character/Barcode



VI-10 Features

- A PC based image processing package bundled with a complete UNIX operating system.
- An interpreted image processing scripting language with "hooks" for incorporating your "C" algorithms.
- Hundreds of existing routines available in our image processing library.
- On-line customer support via internet and phone allowing remote site servicing of your system.
- "Headless" operation (no keyboard or monitor) available for production environments.
- Shared application development and production environments.

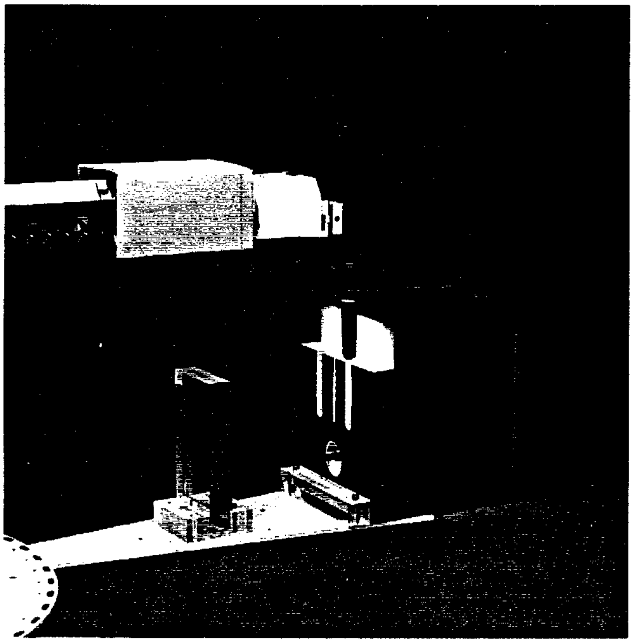
Vision Instruments Inc.
PO Box 130522
Ann Arbor, Michigan
48113-0522

Tel: (313) 994-9260
Fax: (313) 994-8618
E-mail: info@vi.com

Vision Instruments

Vision Instruments Zymate® Compatible Product Information

Liquid Level Detection PySection™



The Liquid Level Detection PySection identifies the location of liquid/solid and liquid/liquid interfaces.

The Liquid Level Detection PySection:

- Reports number of boundary level interfaces.
- Reports absolute position of boundary level interfaces.
- Maintains an image audit trail of containers processed.
- Computes volumetric content of containers.

Used in conjunction with Vision Instruments' Vision Engine technology, the Liquid Level Detection PySection provides quantitative and qualitative data concerning liquid/air, liquid/liquid, and liquid/solid interfaces.

The data returned by this PySection identify the number of boundary level interfaces and their physical locations (in mm) relative to the bottom of the test tube.

These data enable precise and robust sample extraction as well as accurate volumetric computation.

The EasyLab® compatible software includes routines for test tube transfer, reporting the number and location of the layers.

Diagnostic and error detection techniques are used to determine the presence/absence of a sample within a test tube, correctness of the container, and correct positioning of the container.

Additional Vision Engine algorithms are available for other data extraction requirements, including precipitate detection, bar code reading, and color-metric classification.

Vision Instruments Inc.
PO Box 130522
Ann Arbor, Michigan
48113-0522

Tel: (313) 994-9260
Fax: (313) 994-8618
E-mail: info@vi.com

Vision Instruments

Specification:

Compatible Containers		Variety of Containers - See Below
Other PySections Required		General-Purpose or Vibrating Hand
Other Vision Instruments Products Required		VI-10
Physical		2 PySectors
	Height:	8.5″ (22 cm)
	Depth:	22″ (56 cm)
	Width:	2.5″ (6.5 cm)
	Weight:	7 lb (3.2 kg)

Ordering Information:

VIS-1	Vision Inspection Test Tube Station - Includes One VIS Block
VIS-1-B-1	Block 1: 16 x 100 mm Tubes, Capacity 2
VIS-1-B-2	Block 2: 20 x 150 mm Tubes, Capacity 2
VIS-1-B-3	Block 3: 25 x 150 mm Tubes, Capacity 2
VIS-1-B-4	Block 4: 50 ml Glass Centrifuge Tubes, Capacity 2
VI-10	Vision Engine Software and PC-based Platform